Predicting College Football Commitments Using Statistical and Machine Learning

Davy Walker[1,], Jingyi Zheng[2] and Nedret Billor[3]*

[1]Undergraduate Student, Department of Mathematics and Statistics & Department of Computer Science and Software Engineering, Auburn University
[2]Assistant Professor, Department of Mathematics and Statistics, Auburn University
[3]Professor of Statistics, Department of Mathematics and Statistics, Auburn University

Abstract

One of the keys to success when building a college football team is skilled talent. Such talent is in high demand, with finite resources existing with which to sway it. Predicting recruits' decisions before they occur is, accordingly, important for recruiting staff. This knowledge allows them to optimally distribute their resources among players, filling gaps before they occur and recruiting the best players available to the team.

Making this prediction on the individual scale is difficult, however. College applications shape the rest of a player's life, so the pressure on them rises and makes predicting individual decisions a challenge. This study uses statistical analysis enhanced with machine learning techniques such as random forest traversals, nearest prototype classifications, and support vector machines.

This model is validated using a dataset with 30+ features on the 1,200 students Auburn University made an offer to from 2017 to 2021. In this snapshot of high school recruits, we find significant success using a support vector machine for predicting the college a recruit will attend, and using an ensemble of a perceptron, decision tree, nearest neighbor classifier, and quadratic discriminant analysis for predicting if a recruit will attend a specific university.

Key Words: Machine Learning, Statistical Learning, Support Vector Machine, College Football

Introduction

In the United States, the national pastime has shifted. No longer does baseball rule, football has supplanted it. In 2019-2020, Auburn University earned $97M from football. All other sports combined earned $15M over the same period. These sports earn a sixth of what football generates but cost Auburn the same [1].

Winning plays a leading role in this earning. High-value teams – such as Auburn, playing in the Southeastern Conference – earn roughly 3% more money for each win [2]. Success in college football can also increase the quantity of a school's applications by as much as 8% [3]. In highly competitive conferences such as the Big Ten, Big Twelve, and Auburn's own SEC, successful recruiting strategies account for 63% to 80% of a team's overall success and winning [4]. Successfully recruiting high-talent players becomes an important task.

This paper details the creation of predictive models designed to provide football staff with knowledge on how likely a recruit is to say yes to an offer before giving it. These findings are based in algorithms from the scikit-learn Python package [5]. Through this information, staff will be able to better allocate resources when recruiting players. It is not the place of this paper to replace staff, as they will know best what players they want to recruit and what talent can fill gaps on the team. Instead, the data this study provides is meant to serve as a potential aid to staff so that they can divide their recruitment resources more optimally.

Data

These algorithms were trained and tested on the 1,200 recruits Auburn has made an offer to within the last 5 years, utilizing over thirty factors in their decision making process. The study required a list of all students in this category. 247Sports online database filled this niche but did not have an easy method for access [6].

[*] Corresponding author: dcw0036@auburn.edu

As such, a web scraper was built using the Python package Beautiful Soup [7] to gather these players' names, their talent scores on a scale of 0 to 1, their position, their hometown, and what school they eventually ended up enrolling at seen in Table 1. Factors such as height and weight were gathered but were determined to be unhelpful. Also gathered were all the events involving Auburn and other teams a recruit had been involved with, but this data was not used in the process due to being gathered after an offer is made, not before.

Table 1.1 Subsection of the first half of the athlete database. Each athlete is assigned a unique ID and listed by a score to 4 degrees of precision between 0 and 1, the position they play, and where their listed hometown is.

ID	Score	Position	Hometown
46040471	.9059	DT	Lawrenceville, GA
46040514	.9621	WR	Saint Louis, MO
46040540	.8593	RB	Stone Mountain, GA
46040571	.9186	WR	Memphis, TN
46040672	.9928	RB	Tucson, AZ
46040685	.8548	OT	Memphis, TN
46040691	.8851	WR	Centerton, AR
…	…	…	…
46041185	.9420	S	Griffin, GA

Table 1.2 Subsection of the second half of the athlete database. Each athlete is listed by a set of coordinates, height and weight, and where they officially enrolled at.

ID	Coord.	Meas.	Enrollment
46040471	84W 34N	75" 303lbs	Florida
46040514	90W 39N	74" 169lbs	Ohio State
46040540	84W 34N	71" 211lbs	A&M
46040571	89W 35N	75" 206lbs	Arkansas
46040672	111W 32N	72" 200lbs	Ole Miss
46040685	90W 36N	78" 315lbs	LSU
46040691	94W 27N	74" 180lbs	Baylor
…	…	…	…
46041185	84W 33N	72" 200lbs	Auburn

This study identified Auburn's greatest recruiting rivals using this information. The enrollments of each student Auburn sent an offer to were tallied, and the 25 largest were taken as Auburn's recruiting rivals.

From here, the hometown of each student was converted to longitude and latitude through the Geopy Python package [8]. The haversine distance between this longitude and latitude and the longitude and latitude of Auburn and the rivals were taken and stored as a factor. The longitude and latitude of each student was compared to a list of every county in the United States provided by SimpleMaps [9], with the county with the smallest haversine distance to the player being assigned.

Using information from the 2015 US Census, the median household income, population below poverty, population above poverty, population above 25, and population above 25 with education above the high school level were gathered for each student's county [10] seen in Table 2. This was converted into the county's median household income as a fraction of the United States' median household income, the percentage of the county below poverty, and the percentage of the population in the county over twenty-five with education above the high school level, which were taken as the final factors for the machine learning algorithms.

Table 2 Example table from the 2015 US Census data detailing those above and below poverty. Contains the number of people and the margin of error.

Label	Estimate	MoE
Total	308,619,550	±10,903
Income in the past year below poverty level	47,749,043	±280,598
Income in the past year at or above poverty level	260,870,507	±287,381

Methods and Analysis

Taking the athletes and factors defined above, the data was split, randomly selecting one-fifth of the data to serve as tests for the algorithms while the remaining data was used for training. Eighteen algorithms were chosen for testing, with the thought process that even failed algorithms would give added information. Some of these failures did in fact provide interesting results, and are included below, but other algorithms produced such similar results as other algorithms already tested that they have been overlooked for conciseness. Each algorithm was trained on the training set, where they learned the relationships between each feature and corrected themselves based on the official enrollment.

Each algorithm was used for two experiments: predicting which college among the chosen university – Auburn – and its 25 rivals a recruit will attend, and predicting whether a recruit will attend the chosen university. These were measured by two metrics: accuracy, and recall. Accuracy measures the percentage of the time the algorithm predicts a recruit successfully on the testing set. Recall does the same, but only looks at the algorithm's accuracy on recruits who attend the chosen university, as opposed to the pool of all recruits sent an offer by the university in the time frame.

The algorithms we continued with were split into four categories. The first was for those which create a tree of decisions through which the algorithm descended to make decisions Fig. 1. Decision Trees are the most simple of these, doing exactly that. Extra Tree makes a group of Decision Trees, taking information from the entire training set but deciding optimizations randomly in order to fill out multiple different trees. Random Forests also make a group of Decision Trees, but instead each of their Trees looks at a random subset of the data, then optimizes based on that subset. The consensus is that ensemble methods such as Extra Trees and Random Forests will produce better results than Decision Trees but have a tradeoff between one another as Extra Trees have less bias and Random Forests have less variance.

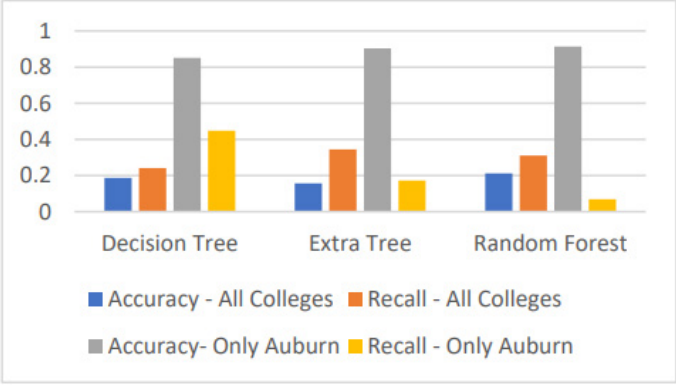


Fig. 1. Tree-based algorithms

The second was for those which plot each training point on n-dimensional space, then determine a prediction based on distance Fig. 2. A Nearest Centroid model does this most explicitly, taking the mean of the coordinates for each point then assigning the point being tested to the prediction of the point with the most similar mean. K-Nearest Neighbors is similar, but instead of looking at means it instead takes the distance between the point being tested and each training point, then takes a vote on the prediction from between the K closest points. We used K=1 and K=3 for this data. Other Ks were tested but ultimately discarded. Some algorithms have a recall of 0, as the algorithm never predicts that a student will attend the chosen university

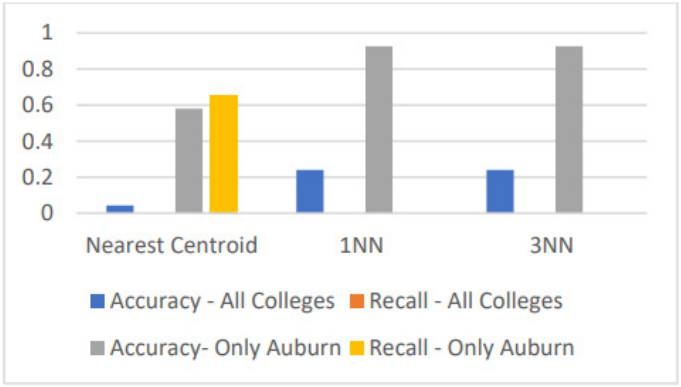


Fig. 2. Nearest-neighbor algorithms

The third was for algorithms which develop linear combinations of variables which discriminate between categories Fig. 3. Linear Discriminant Analysis separates classes by creating a linear combination of the training features optimized so that inputs from a specific class group with similar inputs. Quadratic Discriminant Analysis is similar but creates a quadratic combination. Both require assumptions about the data that our dataset does not match, but they are being included as useful baselines on what to expect in our other models. A Ridge Classifier takes various inputs and converts them all into either -1 or 1 before minimizing the sums of the square of the difference between prediction and target.

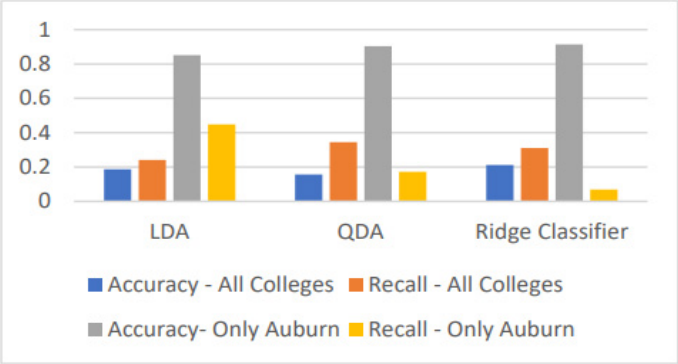


Fig. 3. Statistical learning algorithms.

The final was for algorithms which weigh each factor by an amount the training data calculated, then make a prediction based on that weight Fig. 4. The first of these methods is a Perceptron. A Perceptron is like Linear Discriminant Analysis in that it attempts to separate classes by creating a linear combination of features, but unlike LDA, it is robust and does not require data to be entirely linearly separable to work. A Multi-Layer Perceptron is similar but captures a better idea of the data. Multiple perceptrons are generated and work together to produce an output. This allows for relationships that are not linear to be captured. Support Vector Machines plot each training point in n dimensional space, then form a hyperplane that separates points in one class from another class. In multi-class problems this becomes impossible, so we use One-Versus-Rest methodology as a work around. The SVM will iterate through each class, then create a hyperplane that separates that class from every other class. Predictions are made by looking at this collection.

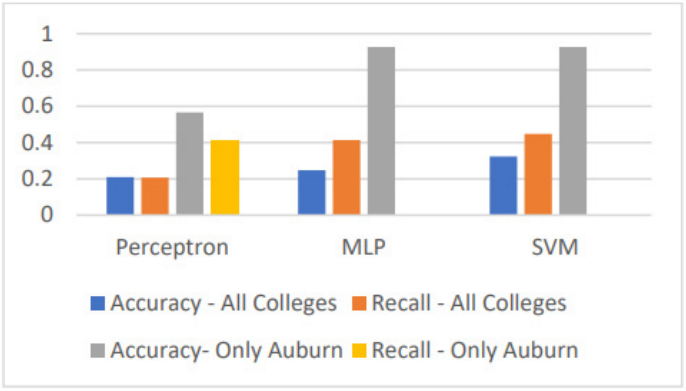


Fig. 4. Perceptron-based algorithms.

Looking only at accuracy and recall when guessing on all colleges, the Ridge Classifier and Support Vector Classifier (SVC) give the best results at 28.7% and 32.3% accuracy respectively. The Ridge Classifier has only a moderate recall at 27.6%, which means that when only looking at the subsection of students who committed to Auburn, it doesn't perform nearly as well. This implies that its prediction strategy leans towards guessing the biggest category rather than the best. The Multi-Layer Perceptron (MLP) and the SVC, meanwhile, have the highest recall when guessing every college at 41.4% and 44.8% recall respectively. This means that rather than just guessing a specific single category repeatedly, they are actually engaging with the material and making an informed estimation on each recruit.

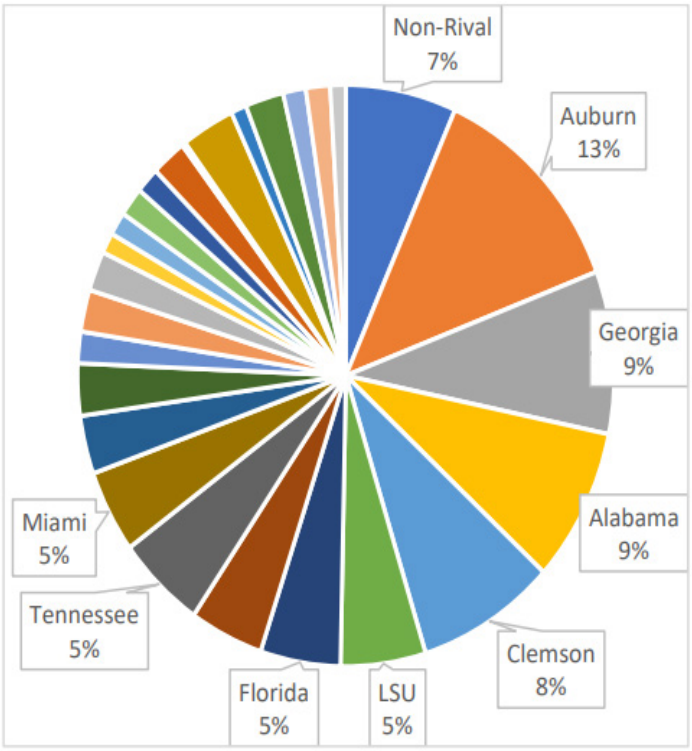


Fig. 5. Example probability map generated by the SVM for making a prediction on which college a specific recruit will attend.

If we instead look at accuracy when guessing Auburn attendance, the results are quite different. This data is lopsided, and many algorithms guess the larger category nearly every time. Quadratic Discriminant Analysis (QDA) performs interestingly on this dataset, almost always guessing a student is attending Auburn instead. The best performer for recall is the Nearest Centroid Classifier (NCC) at 62% accuracy, though the Decision Tree Classifier and Perceptron Classifier have decent results at

45% and 41% respectively.

Looking only at these algorithms as they successfully guess Auburn students, Decision Tree has by far the best accuracy at 85%, while NCC and Perceptron top out at 57%. As seen in Fig. 1-4 various other algorithms have higher accuracy, but all algorithms with recall below 25% were deemed unsuitable to our purposes. The goal of these algorithms is to guess where students will go successfully, rather than just achieve the highest number. If an algorithm tells a coach no on every single student, then even if that no is generally very accurate, it is useless. Taking the successful Decision Tree, NCC, and Perceptron, we'll make an ensemble algorithm which votes between the results of the component algorithms to produce a result.

Voting between Tree, NCC, and Perceptron gives the highest overall result of these votes at 79%, with 55% recall. Voting between Tree, NCC, and QDA meanwhile gives the best recall at 76% and gets 58% accuracy on the entire set. Voting between all four of Tree, NCC, Perceptron, and QDA gives what this study believes to be the most useful result, at 78% overall accuracy with 62% recall. A sample vote is shown in Table 3.

Table 3 Example results for the same recruit as in Fig. 5. Rather than predicting which college the recruit will attend, the algorithms are instead predicting if a recruit will go to the chosen university. Here, even though two of the models in the vote got the answer wrong, the ensemble predicted the result successfully.

Model	Result
Decision Tree	Auburn
Nearest Centroid	Not Auburn
Perceptron	Not Auburn
Quadratic Discriminant Analysis	Auburn
Vote	Auburn
Ground Truth	Auburn

Conclusions

The most impressive of our results was in predicting where a given recruit will attend college. At roughly 33%, this is a very useful metric primarily due to how few sources it uses. Most predictors rely on getting to know a candidate, look at where they visit, and only make a guess after all the offer letters have been given. Our predictor can instead be run at any point in an athlete's decision making process, as the only time reliant feature that might throw off a prediction is an athlete moving, moving position, or having a sharp increase or decrease in skill – all things that are very rare for the 4 and 5 star recruits top schools are looking at.

The goal of this study, however, is to decrease uncertainty for coaches in the early days of recruitment when they are trying to decide who they want to invest time and resources in, which has been accomplished. At nearly 80% accuracy on if a given athlete will attend Auburn, coaches will have a good idea on if their efforts are worth it long before any effort is actually put in.

With the creation of the transfer portal, athletes have more agency in their college sports career than ever before. As such, it is the hope of this study to begin shifting the research narrative away from decisions based on teams and towards decisions based on players. At the end of the day people should be and are the most important part of any endeavor, sports included, and shaping our research around what those people want to do as opposed to what others want to do with them is vital.

Moving to the future, this model should be validated on freshmen, with the database receiving updates every season. More factors from the census can also be gathered, further refining the algorithms as they have more information to input. With that said, the model will likely never be perfect. Choosing what school to attend is a deeply personal decision, and one which no one can fully predict other than the athletes themselves.

References

[1] Staff, Sportico. "Sportico's Intercollegiate Finance Database." Sportico.com, Sportico.com, Retrieved May 24, 2022 from https://www.sportico.com/business/commerce/2021/co llege-sports-finances-database-intercollegiate 1234646029/.

[2] Chung, Doug J. "How Much Is a Win Worth? an Application to Intercollegiate Athletics." Management Science, vol. 63, no. 2, 2017, pp. 548–565., Retrieved May 24, 2022 from https://doi.org/10.1287/mnsc.2015.2337.

[3] Pope, Devin G., and Jaren C. Pope. "The Impact of

College Sports Success on the Quantity and Quality of Student Applications." Southern Economic Journal, vol. 75, no. 3, 2009, pp. 750–780., Retrieved May 24, 2022 from https://doi.org/10.1002/j.2325- 8012.2009.tb00930.x.

[4] Caro, Cary A. "College Football Success: The Relationship between Recruiting and Winning." International Journal of Sports Science & Coaching, vol. 7, no. 1, 2012, pp. 139–152., Retrieved January 11, 2022 from https://doi.org/10.1260/1747-9541.7.1.139.

[5] "Scikit Learn Documentation." Scikit, Retrieved December 11, 2021 from https://scikit-learn.org/stable/.

[6] MarshallVIP, 9 hours agoDotPhillip, et al. "Aubrnundercover Home - Auburn Tigers Football & Recruiting." 247Sports, Retrieved December 29, 2021 from https://247sports.com/college/auburn/.

[7] "Beautiful Soup Documentation¶." Beautiful Soup Documentation - Beautiful Soup 4.9.0 Documentation, Retrieved December 29, 2021 from https://www.crummy.com/software/BeautifulSoup/bs4 /doc/.

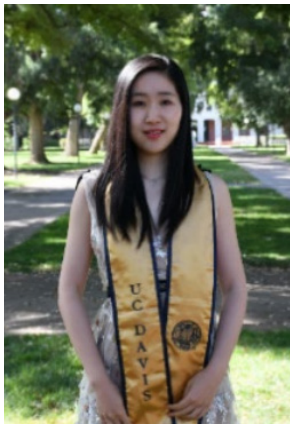
[8] "Welcome to GeoPy's Documentation!" – GeoPy 2.3.0 Documentation, Retrieved April 9, 2022 from https://geopy.readthedocs.io/en/stable/.

[9] "United States Counties Database." Simplemaps, Retrieved April 9, 2022 from https://simplemaps.com/data/us-counties.

[10] Bureau, U.S. Census. "Explore Census Data." Explore Census Data, Retrieved April 9, 2022 from https://data.census.gov/.

Authors Biography

Davy Walker is a senior-year student pursuing a B.S. in Applied Discrete Mathematics and a B.S. in Computer Science concurrently at Auburn University. They have played a key role in problem conception, code generation, and statistical analysis. Upon graduation they intend to pursue their M.S. in Data Science/ Data Engineering from Auburn University.



Dr. Jingyi Zheng is an Assistant Professor in the Department of Mathematics and Statistics at Auburn University. Her research interests are in areas of Data Science, Machine Learning, and Data-driven computing.

Dr. Nedret Billor is a Professor of Statistics in the Department of Mathematics and Statistics at Auburn University. Her primary interests include robust multivariate data analysis, robust functional data analysis, outlier detection